

# Mastering the game of Go with deep neural networks and tree search

David Silver<sup>1\*</sup>, Aja Huang<sup>1\*</sup>, Chris J. Maddison<sup>1</sup>, Arthur Guez<sup>1</sup>, Laurent Sifre<sup>1</sup>, George van den Driessche<sup>1</sup>, Julian Schrittwieser<sup>1</sup>, Ioannis Antonoglou<sup>1</sup>, Veda Panneershelvam<sup>1</sup>, Marc Lanctot<sup>1</sup>, Sander Dieleman<sup>1</sup>, Dominik Grewe<sup>1</sup>, John Nham<sup>2</sup>, Nal Kalchbrenner<sup>1</sup>, Ilya Sutskever<sup>2</sup>, Timothy Lillicrap<sup>1</sup>, Madeleine Leach<sup>1</sup>, Koray Kavukcuoglu<sup>1</sup>, Thore Graepel<sup>1</sup> & Demis Hassabis<sup>1</sup>

# AlphaGo

Online/Offline Pipeline

Offline

# Machine Learning:

## Convolutional Neural Networks

Online

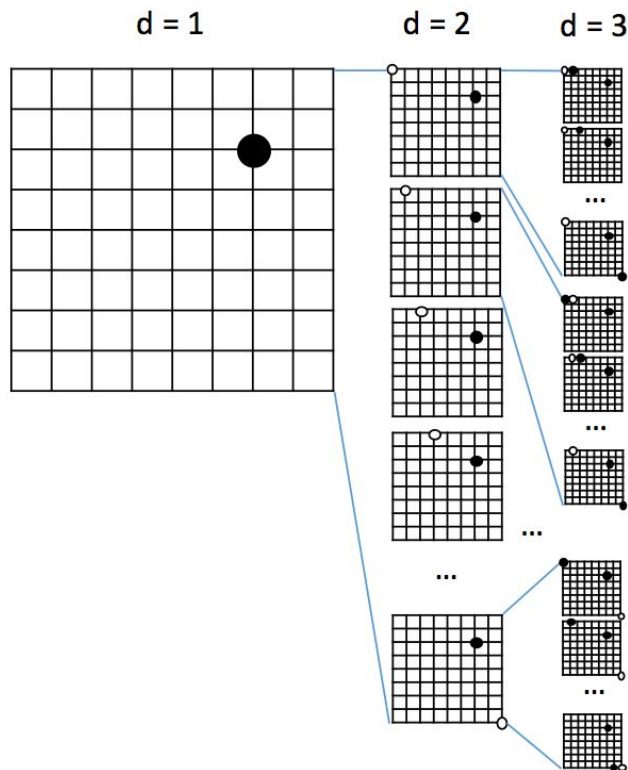
# Search:

## Monte Carlo Tree Search

# Background 1

## Convolutional Neural Networks

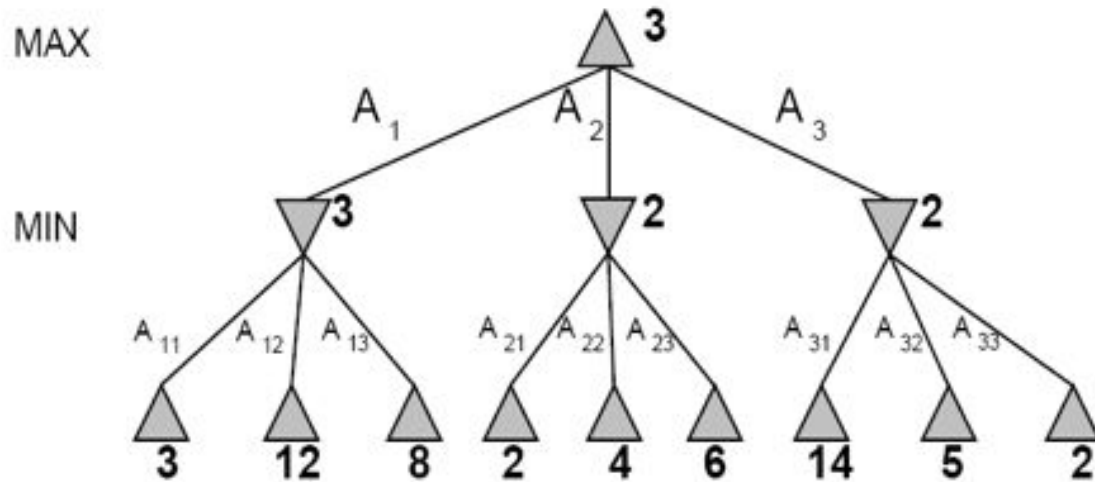
# Idea 1: Tree Search with pruning & heuristic



- Minimax algorithm
- Alpha-beta pruning
- History heuristic
- Killer heuristic
- ...

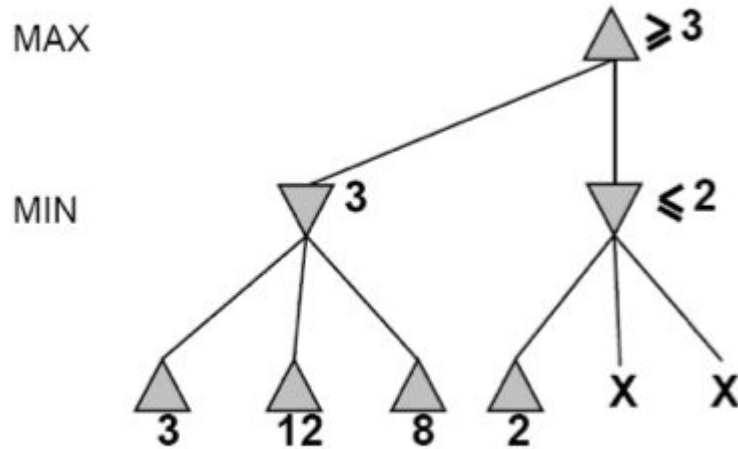
# Minimax algorithm with alpha-beta pruning

2 PLY GAME

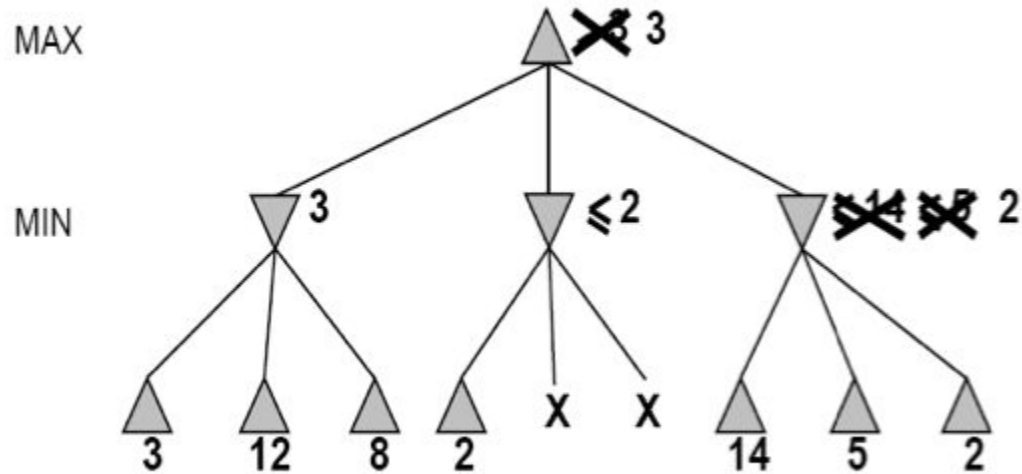


# Minimax algorithm with alpha-beta pruning

3x

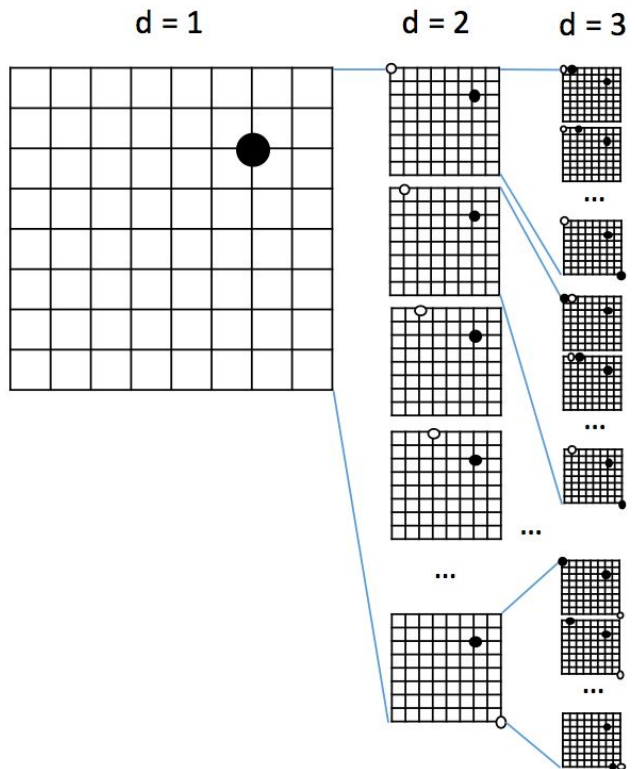


# Minimax algorithm with alpha-beta pruning



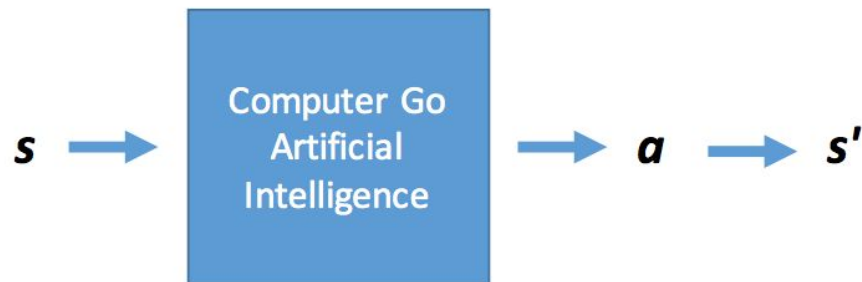
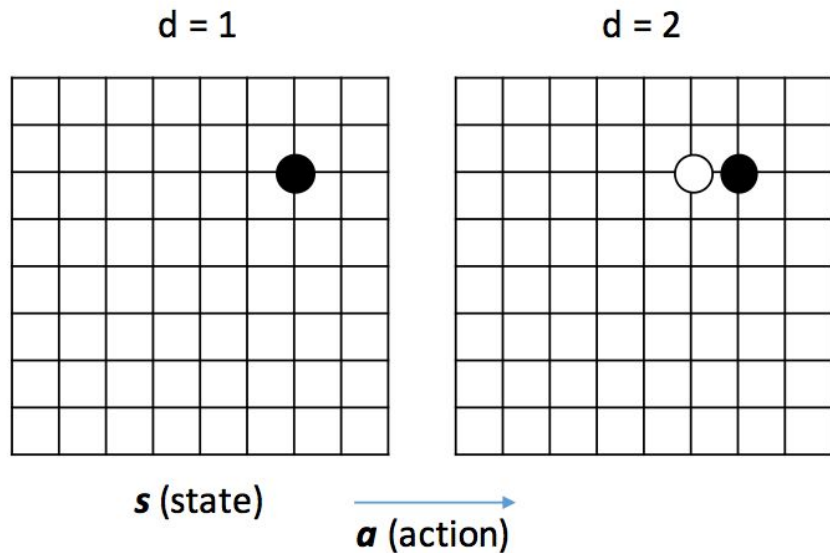


# Idea 1: Tree Search with pruning & heuristic



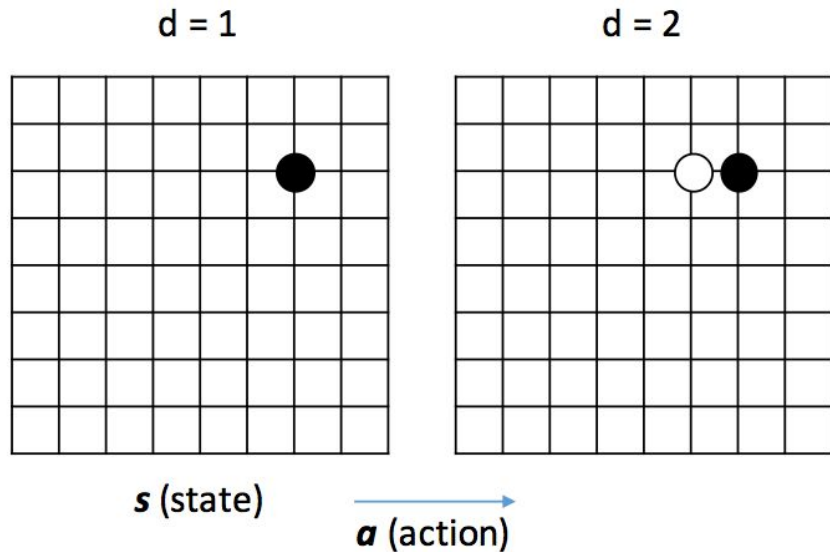
- Before 2002
- Monte Carlo 2002-2006
- MCTS 2006-

## Idea 2: Predict without Tree Search?

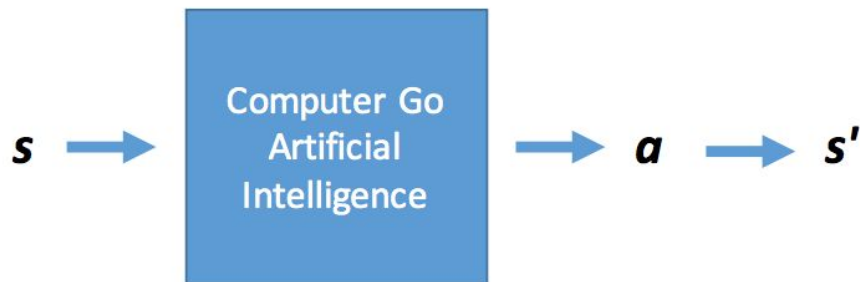


Given  $s$ , pick the best  $a$

## Idea 2: Predict without Tree Search?

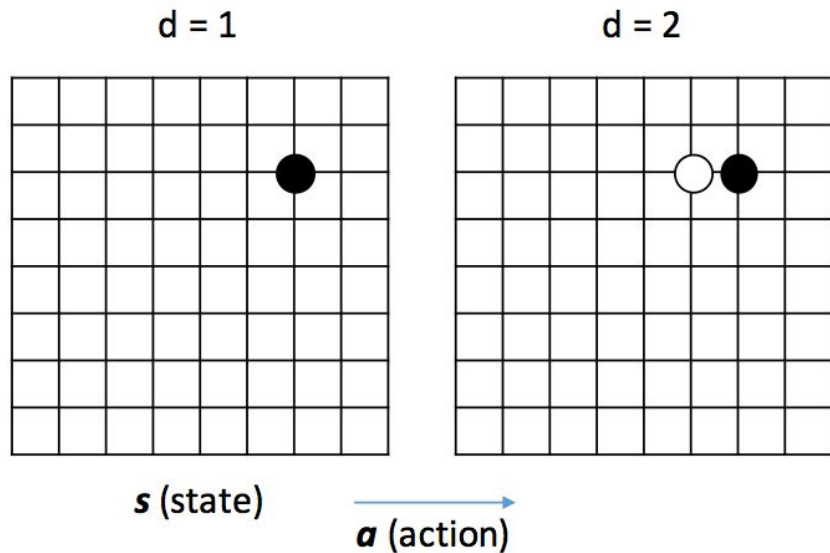


Given  $s$ , pick the best  $a$

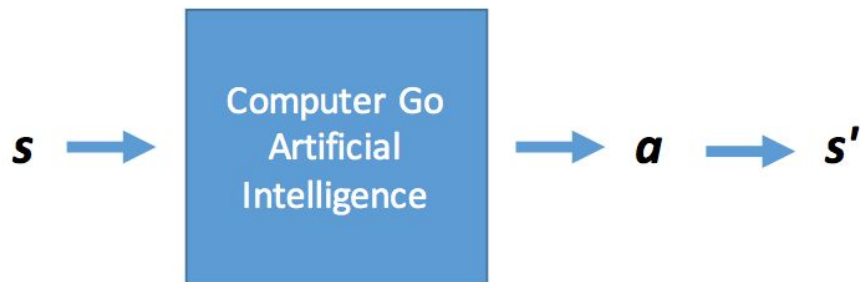


**“Using no search at all, the RL policy network won 85% of games against Pachi.”**

## Idea 2: **Predict** without Tree Search?

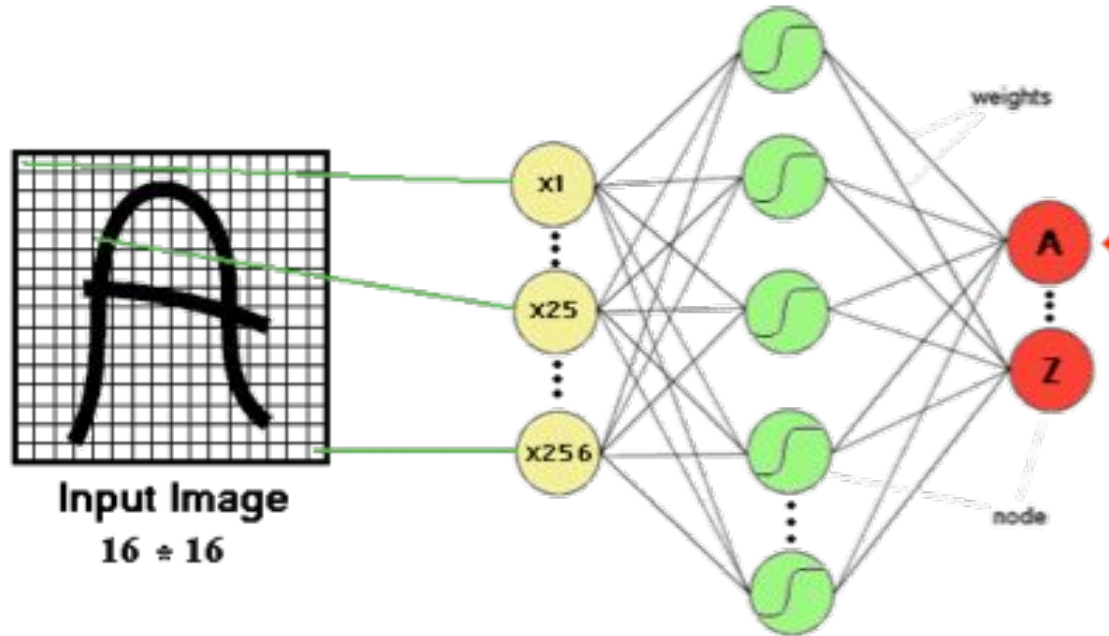


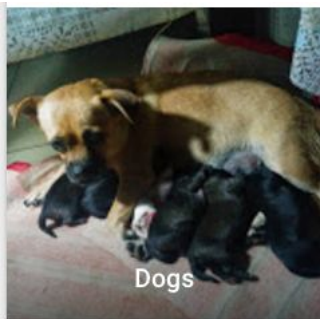
Given  $s$ , pick the best  $a$



**“Using no search at all, the RL policy network won 85% of games against Pachi.”**

# Neural Network as a Classifier





Dogs



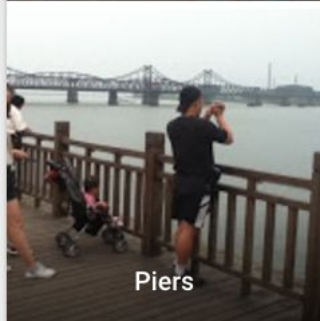
Stadiums



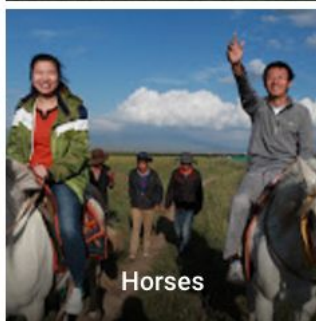
Playgrounds



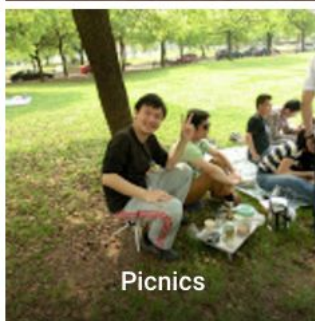
Cliffs



Piers



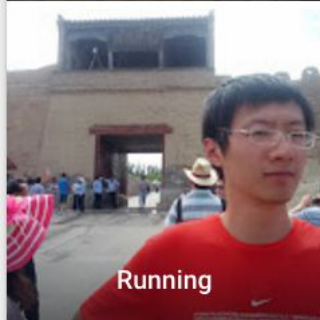
Horses



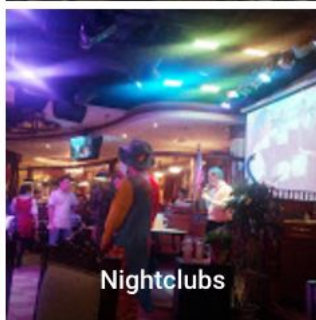
Picnics



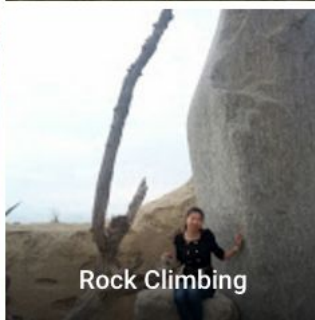
Birds



Running



Nightclubs

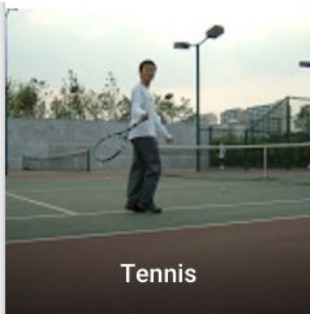


Rock Climbing



Parks

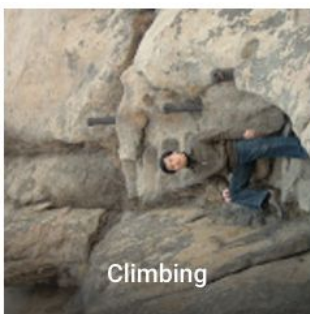




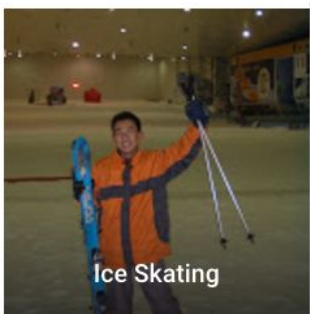
Tennis



Basketball



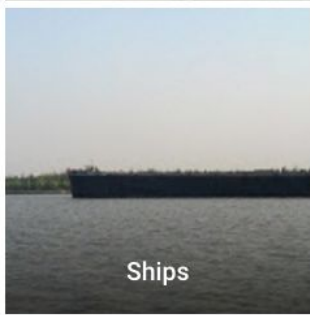
Climbing



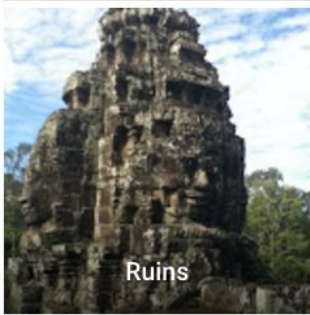
Ice Skating



Table Tennis



Ships



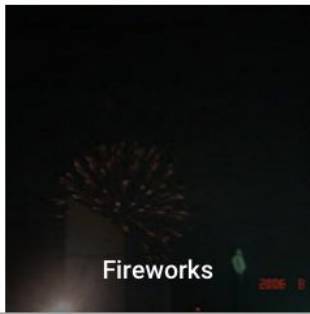
Ruins



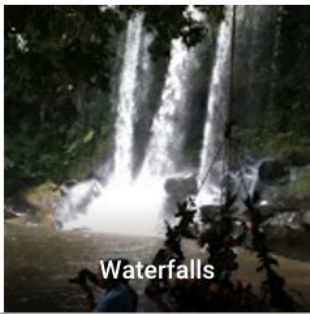
Deserts



Birthday



Fireworks



Waterfalls



Dunes



Forests



Jul 21, 2013



Jul 20, 2013



May 31, 2013



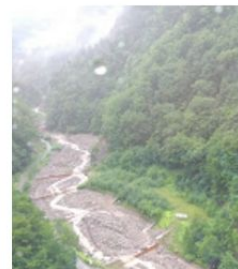
Feb 12, 2013



Oct 1, 2012



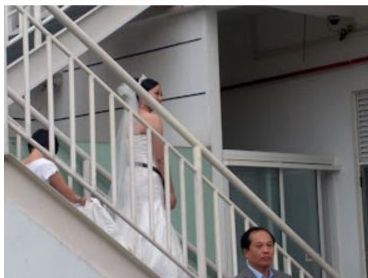
Aug 10, 2012



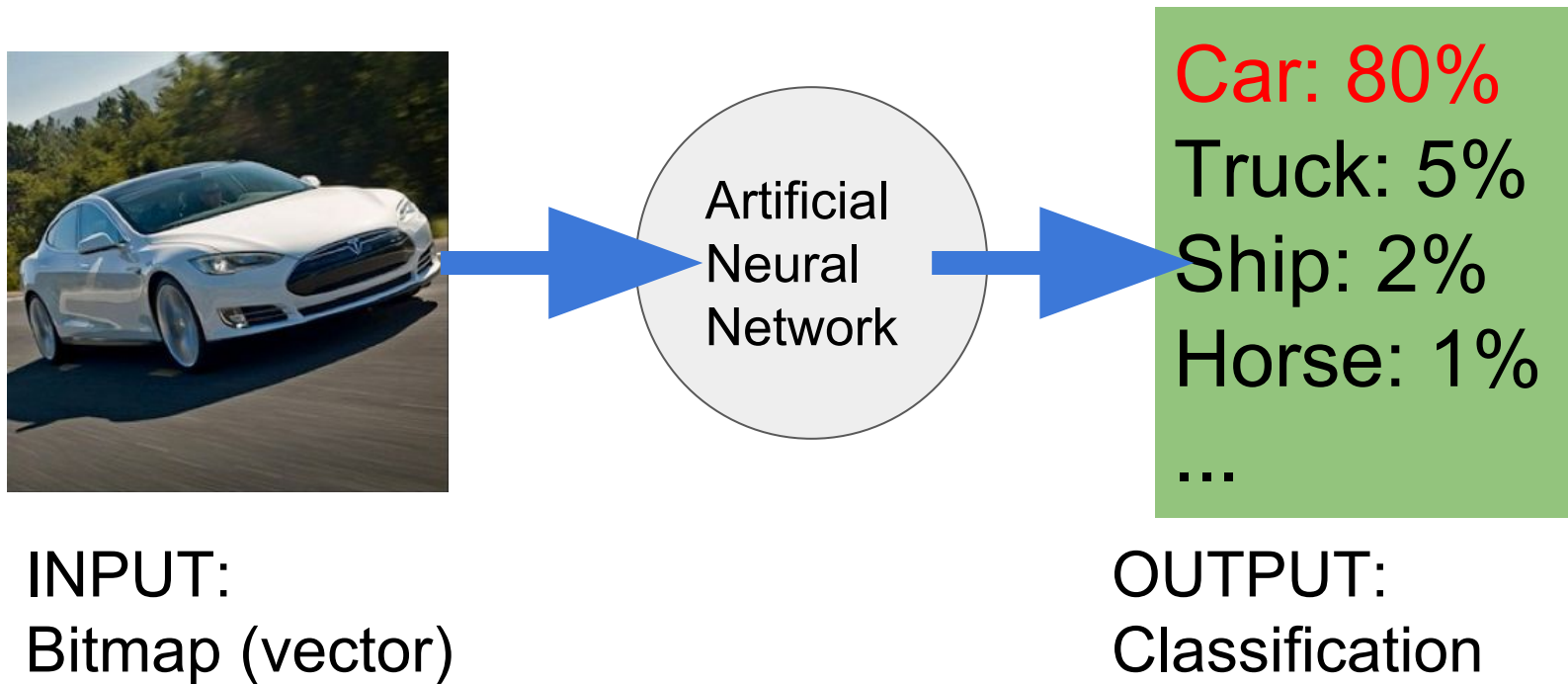




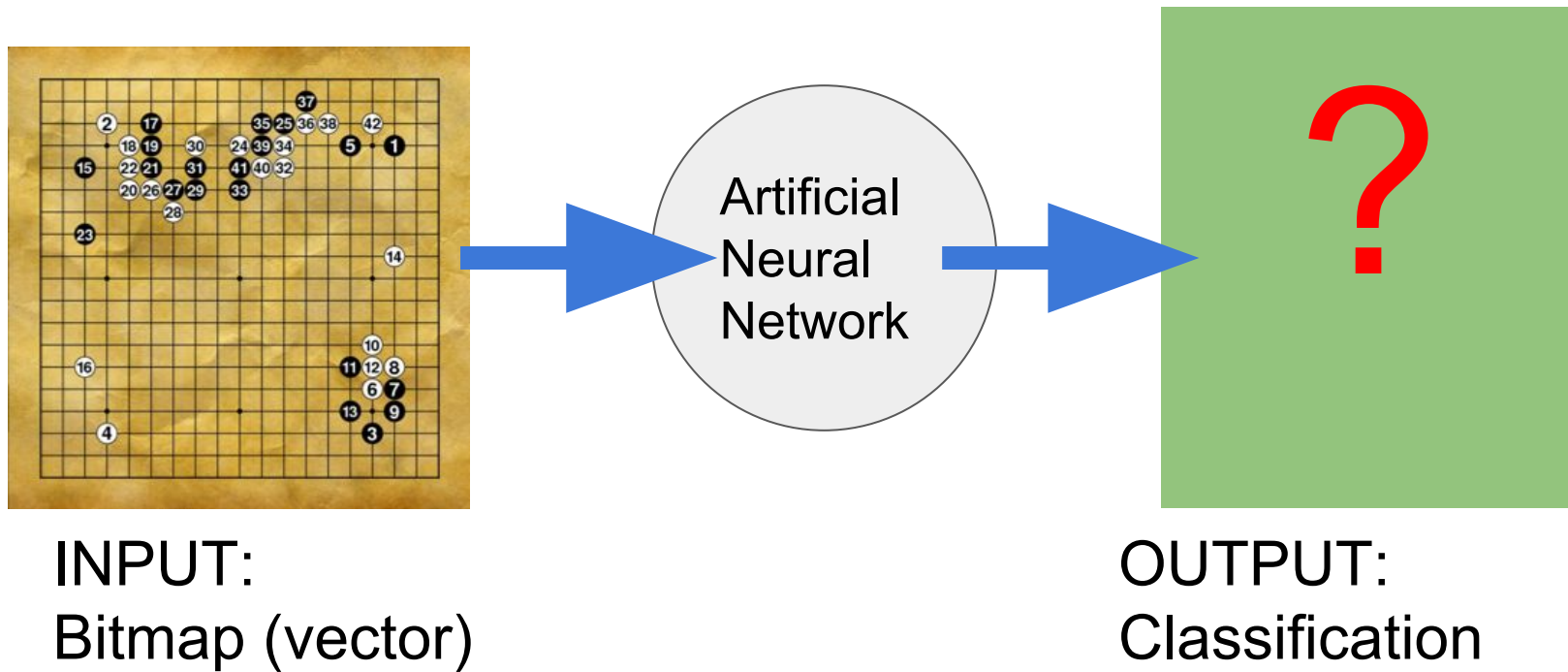
Wedding



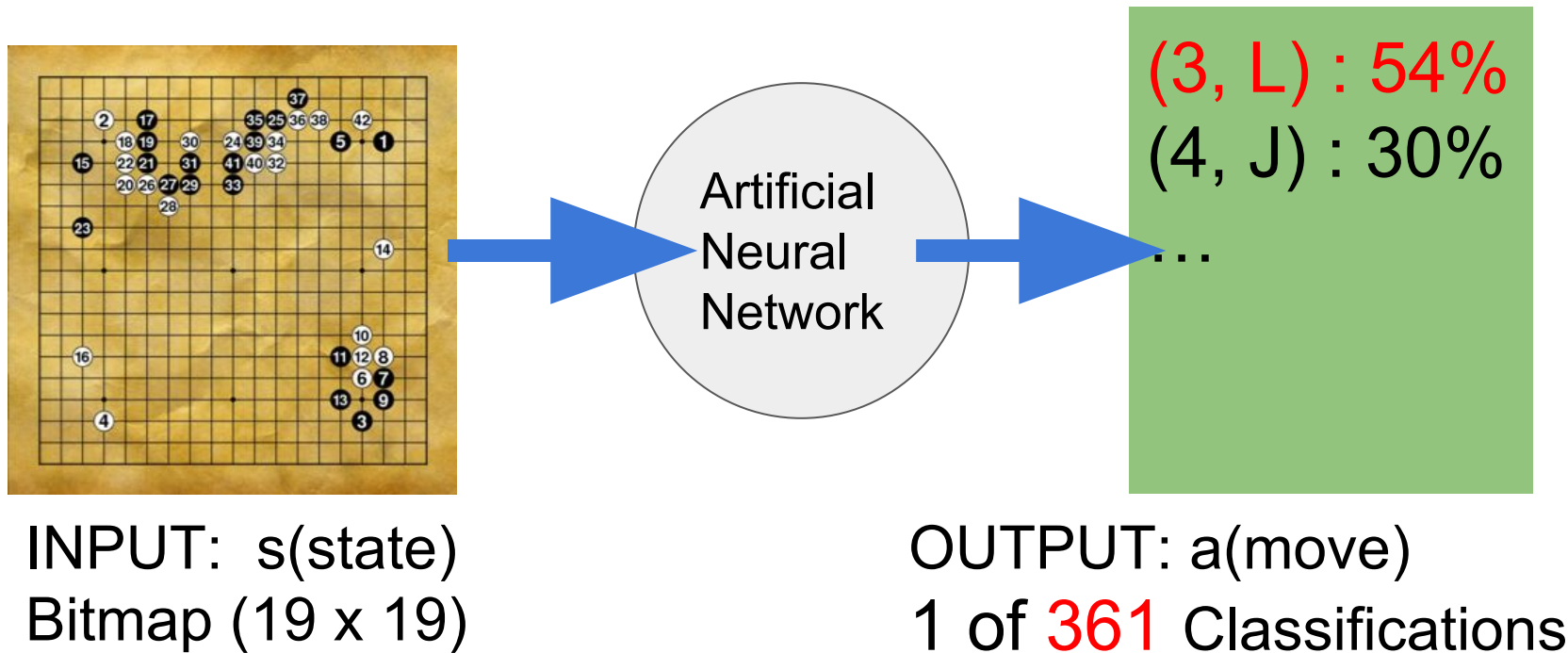
# What does Neural Network do?



# What does Neural Network do?



# What does Neural Network do?



# Convolutional Neural Networks (LeNet, CNN)

- Introduced by Yann LeCun, 1989
- Algorithm
  - **Convolve 1** - Feature extraction
  - **Subsample 1** - reduce the spatial resolution
  - **Convolve 2 ...**
  - **Subsample 2 ...**
  - ...



# Convolve

|                 |                 |                 |   |   |
|-----------------|-----------------|-----------------|---|---|
| 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 0 | 0 |
| 0 <sub>x0</sub> | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 | 0 |
| 0 <sub>x1</sub> | 0 <sub>x0</sub> | 1 <sub>x1</sub> | 1 | 1 |
| 0               | 0               | 1               | 1 | 0 |
| 0               | 1               | 1               | 0 | 0 |

Image

|   |  |  |
|---|--|--|
| 4 |  |  |
|   |  |  |
|   |  |  |

Convolved  
Feature



# Convolve

|   |                 |                 |                 |   |
|---|-----------------|-----------------|-----------------|---|
| 1 | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 0 <sub>x1</sub> | 0 |
| 0 | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 0 |
| 0 | 0 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 1 |
| 0 | 0               | 1               | 1               | 0 |
| 0 | 1               | 1               | 0               | 0 |

Image

|   |   |  |
|---|---|--|
| 4 | 3 |  |
|   |   |  |
|   |   |  |

Convolved  
Feature

# Convolve

|   |   |                 |                 |                 |
|---|---|-----------------|-----------------|-----------------|
| 1 | 1 | 1 <sub>x1</sub> | 0 <sub>x0</sub> | 0 <sub>x1</sub> |
| 0 | 1 | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 0 <sub>x0</sub> |
| 0 | 0 | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> |
| 0 | 0 | 1               | 1               | 0               |
| 0 | 1 | 1               | 0               | 0               |

Image

|   |   |   |
|---|---|---|
| 4 | 3 | 4 |
|   |   |   |
|   |   |   |

Convolved  
Feature



# Convolve

|                 |                 |                 |   |   |
|-----------------|-----------------|-----------------|---|---|
| 1               | 1               | 1               | 0 | 0 |
| 0 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 1 | 0 |
| 0 <sub>x0</sub> | 0 <sub>x1</sub> | 1 <sub>x0</sub> | 1 | 1 |
| 0 <sub>x1</sub> | 0 <sub>x0</sub> | 1 <sub>x1</sub> | 1 | 0 |
| 0               | 1               | 1               | 0 | 0 |

Image

|   |   |   |
|---|---|---|
| 4 | 3 | 4 |
| 2 |   |   |
|   |   |   |

Convolved  
Feature

# Convolve

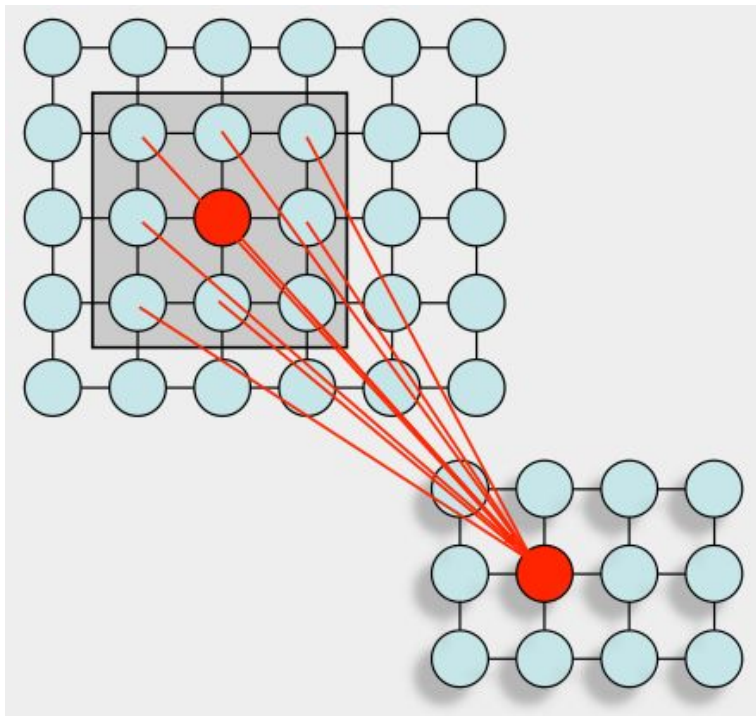
|   |   |                 |                 |                 |
|---|---|-----------------|-----------------|-----------------|
| 1 | 1 | 1               | 0               | 0               |
| 0 | 1 | 1               | 1               | 0               |
| 0 | 0 | 1 <sub>x1</sub> | 1 <sub>x0</sub> | 1 <sub>x1</sub> |
| 0 | 0 | 1 <sub>x0</sub> | 1 <sub>x1</sub> | 0 <sub>x0</sub> |
| 0 | 1 | 1 <sub>x1</sub> | 0 <sub>x0</sub> | 0 <sub>x1</sub> |

Image

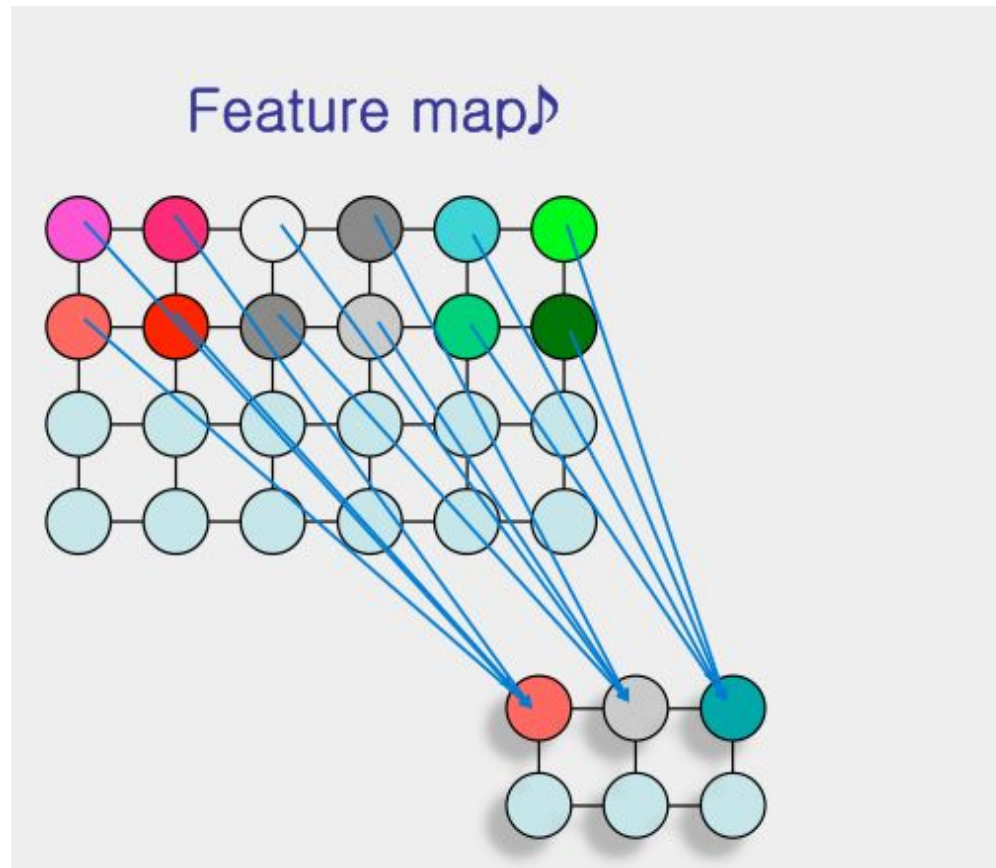
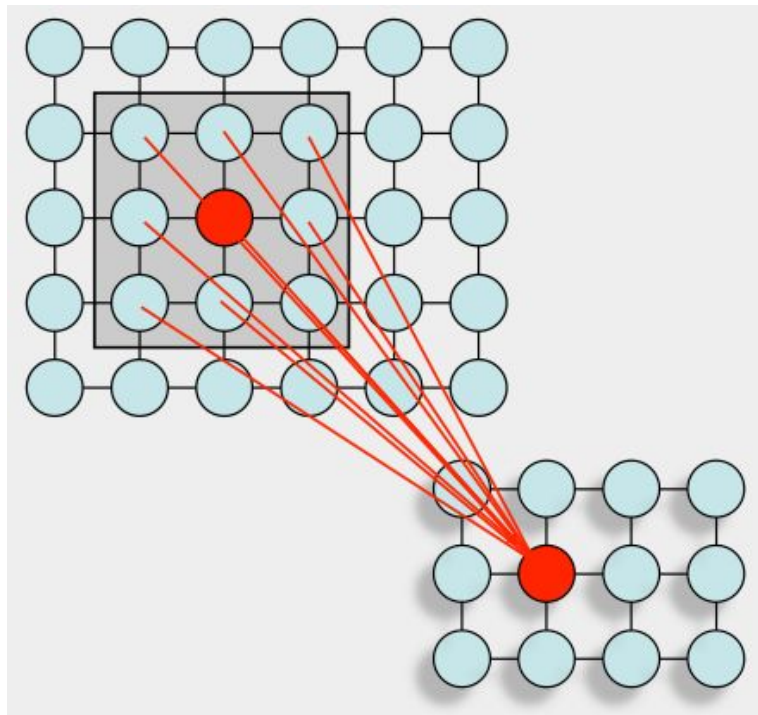
|   |   |   |
|---|---|---|
| 4 | 3 | 4 |
| 2 | 4 | 3 |
| 2 | 3 | 4 |

Convolved  
Feature

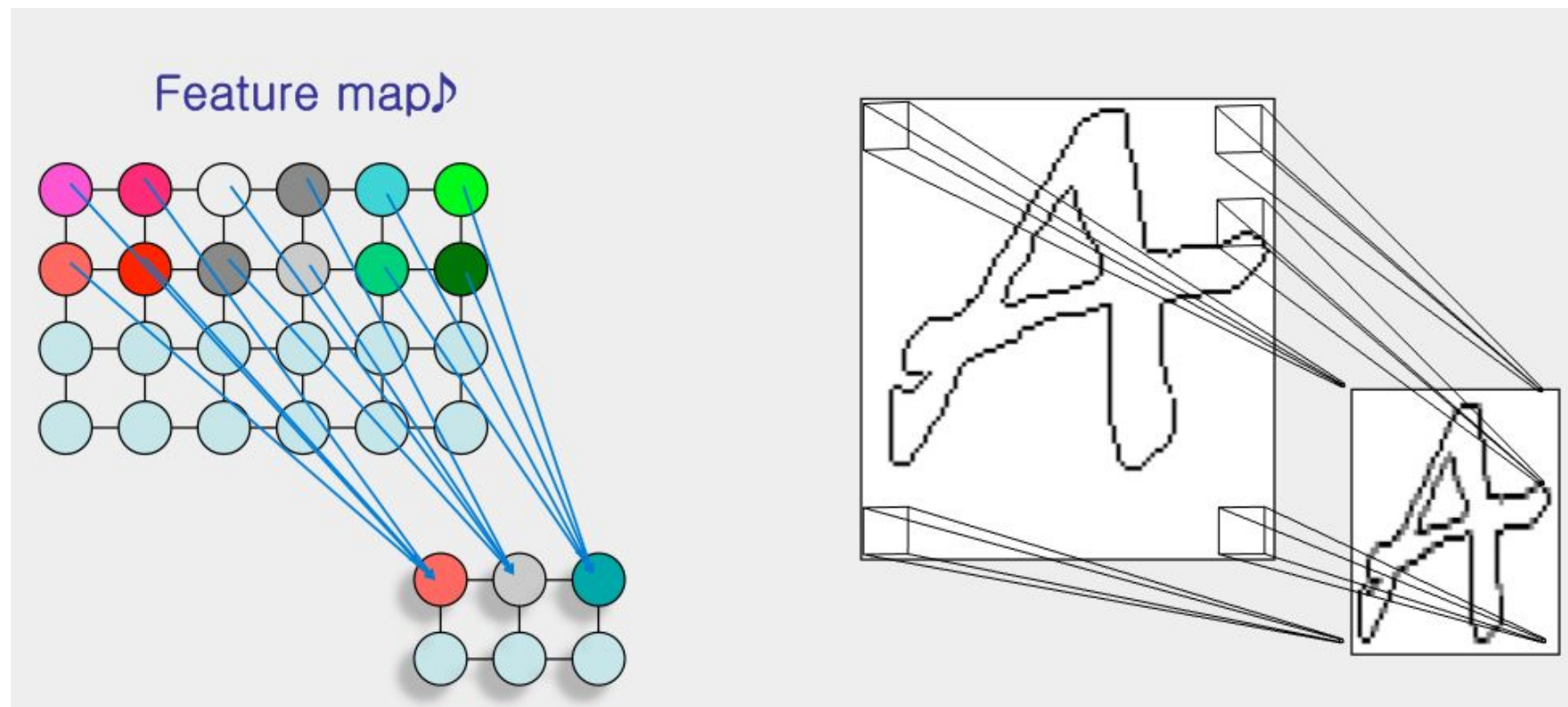
# Convolve vs. Subsample



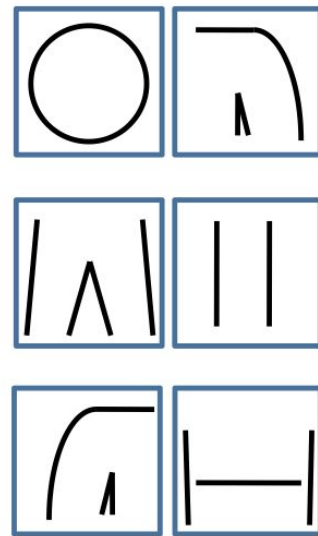
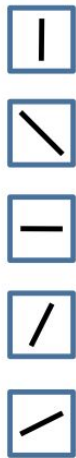
# Convolve vs. Subsample



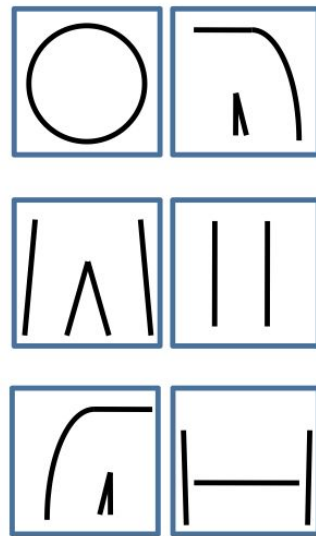
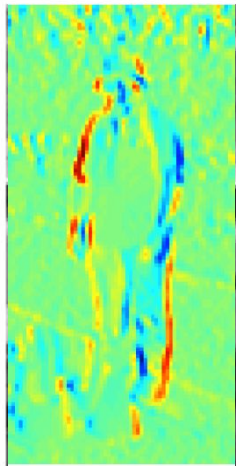
# Subsample



# Subsample



# Subsample

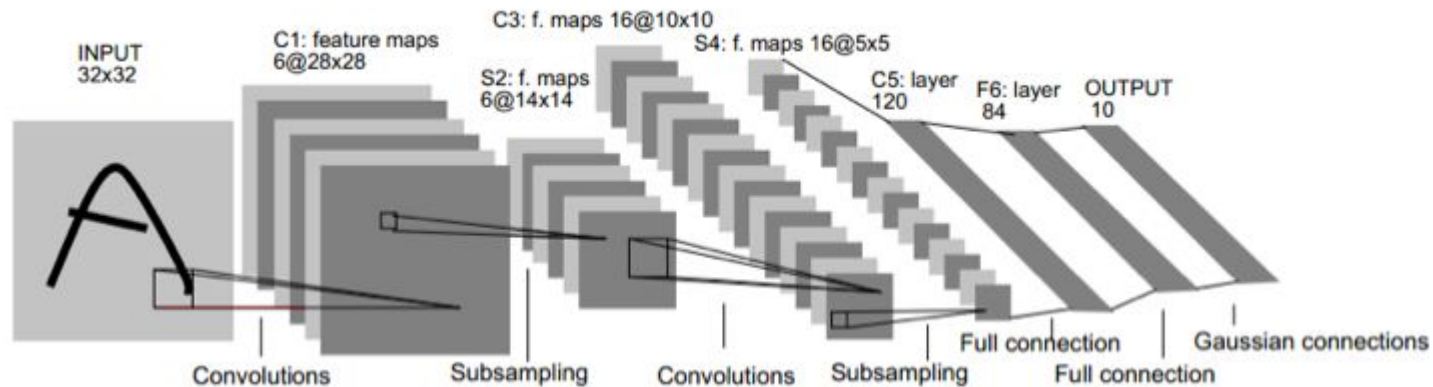


## CNN - other concepts

- Full connect layer
- Non-linearity
- Back Propagation

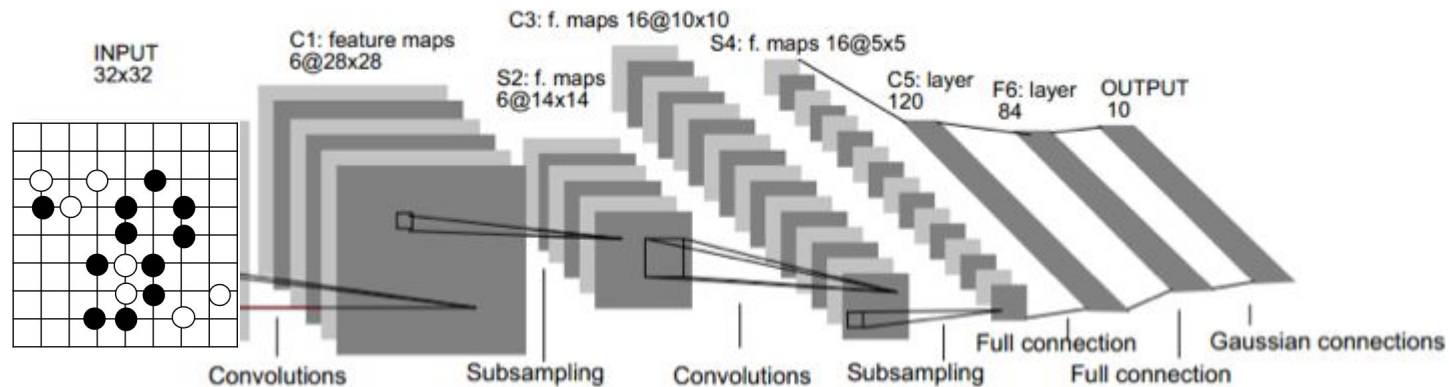


# LeNet-5



|         | L1: C   | L2: S   | L3: C    | L4: S  | L5: FC |
|---------|---------|---------|----------|--------|--------|
| INPUT   | 32x32   |         |          |        |        |
| FEATURE | 6       |         |          |        |        |
| OUTPUT  | 6@28x28 | 6@14x14 | 16@10x10 | 16@5x5 | 120    |

# AlphaGo



LAYERS: 13

INPUT: 23x23 (19+4)

FEATURE: 48

# AlphaGo

Offline Pipeline

Offline

# Machine Learning:

## Convolutional Neural Networks

Online

# Search:

## Monte Carlo Tree Search

# Machine Learning:

## Convolutional Neural Networks

- 3 policy network
  - rollout, supervised, reinforcement
- 1 value network

# Search:

## Monte Carlo Tree Search

Offline

## STEP 1: supervised learning

Expert Positions



Small  
feature  
patterns

PN1:  
Rollout



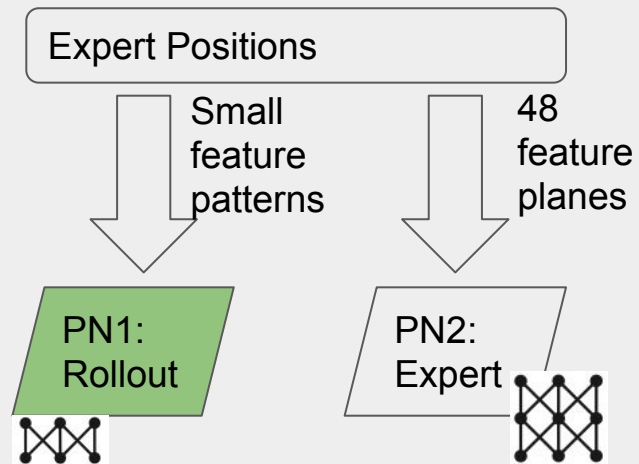
Online

# Search:

## Monte Carlo Tree Search

Offline

## STEP 1: supervised learning



Online

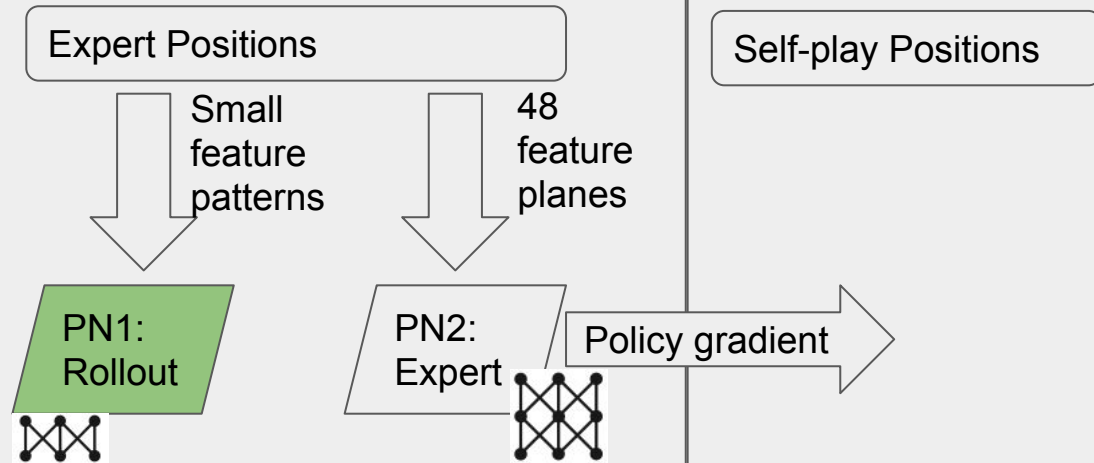
# Search:

## Monte Carlo Tree Search

Offline

## STEP 1: supervised learning

## S2: reinforcement learning



Online

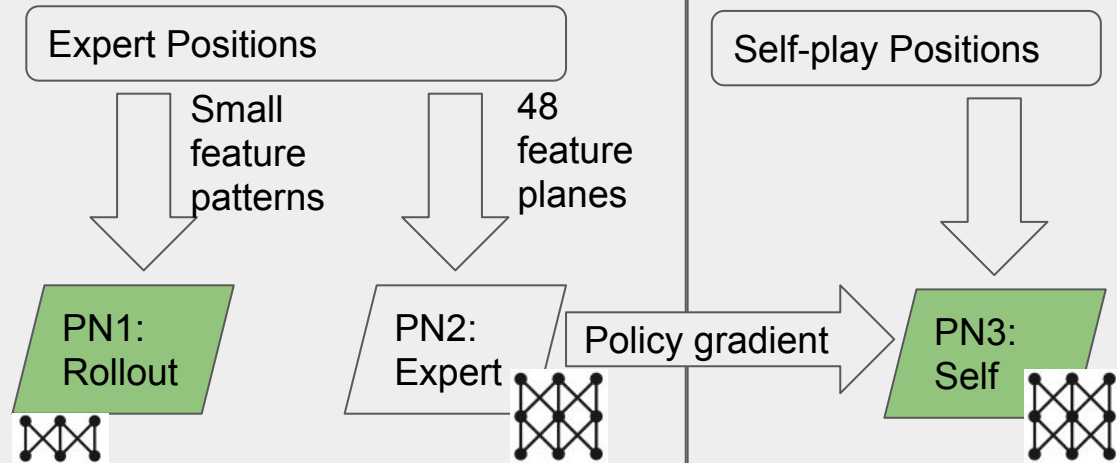
Search:  
Monte Carlo Tree Search



Offline

## STEP 1: supervised learning

## S2: reinforcement learning



Online

# Search:

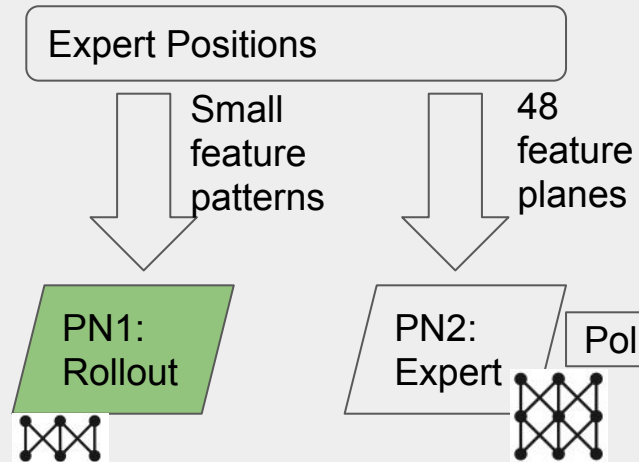
## Monte Carlo Tree Search

Offline

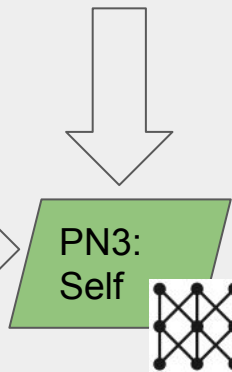
## STEP 1: supervised learning

## S2: reinforcement learning

## STEP3: value networks



Self-play Positions



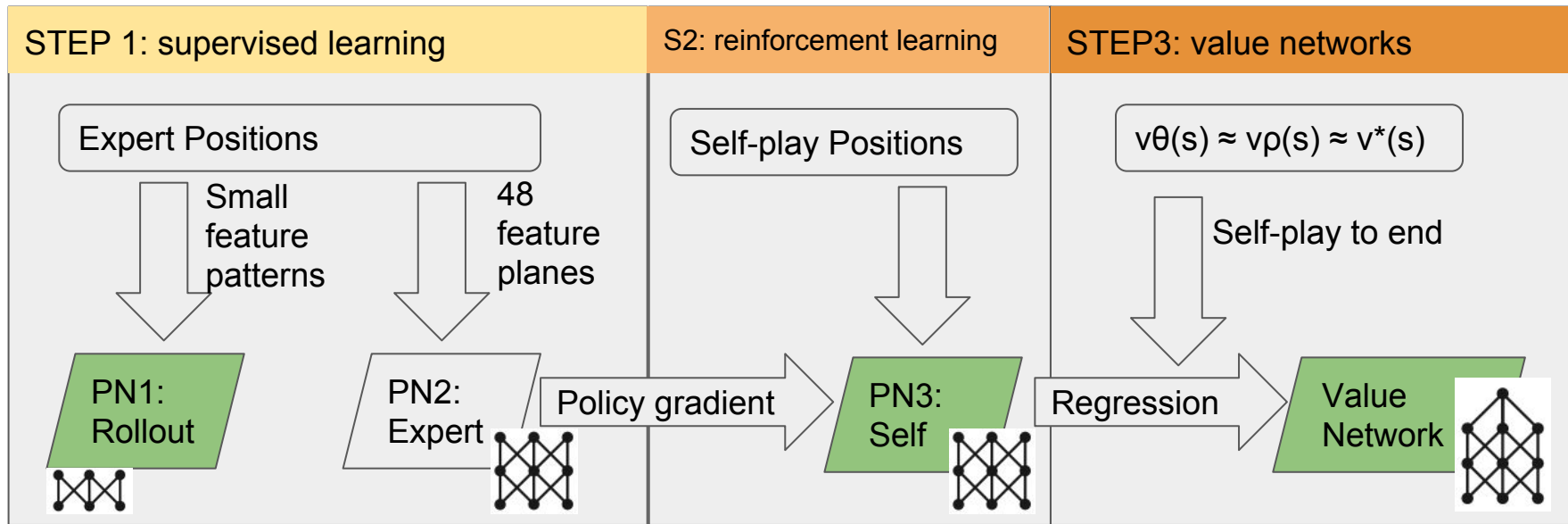
$$v_{\theta}(s) \approx v_{\rho}(s) \approx v^*(s)$$



Online

Search:  
Monte Carlo Tree Search

Offline



Online

Search:  
Monte Carlo Tree Search

# Evaluation 1: Won 85% to Pachi

11% => 85%

Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play.

# Background 2

## Monte Carlo Tree Search

# Random Algorithms

- Las Vegas
- Monte Carlo

# Random Algorithms: Las Vegas



- Find the right key

# Random Algorithms: Las Vegas



- Find the right key
- Always correct, or fail



# Random Algorithms: Monte Carlo



- Find the biggest apple

# Random Algorithms: Monte Carlo



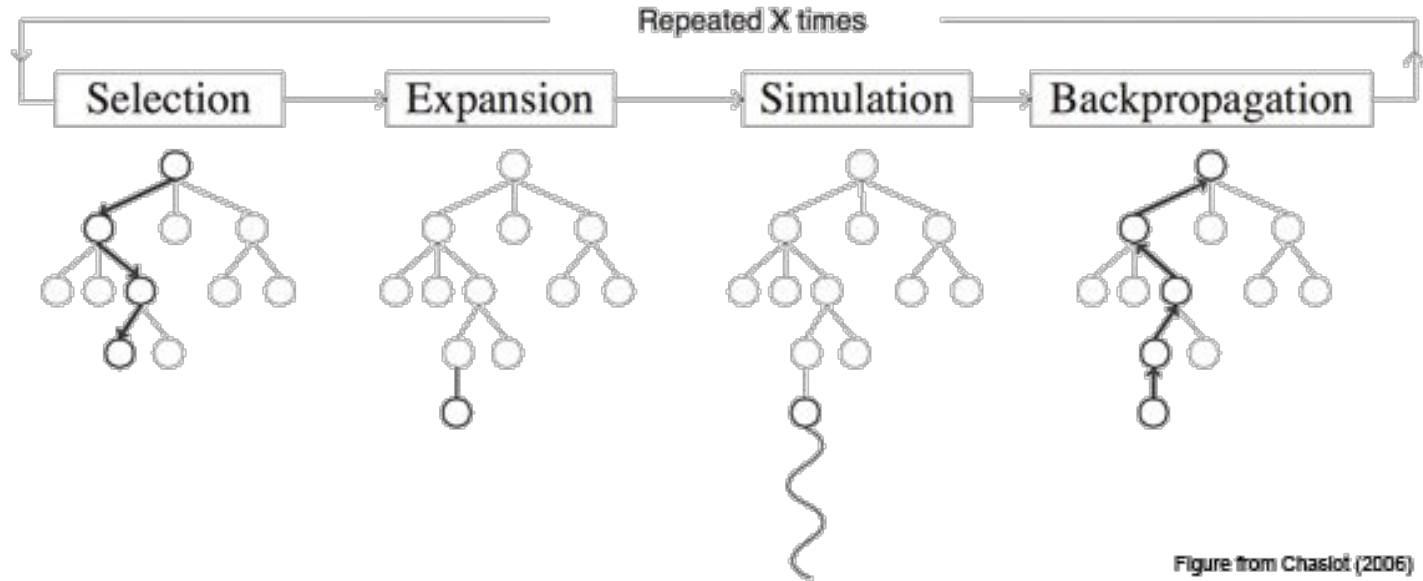
- Find the biggest apple
- May not correct, never fail

# Random Algorithms: Monte Carlo



- Find the biggest apple
- May not correct, never fail
- Algorithm
  - SELECT
  - SELECT Another
  - EVALUATION
  - UPDATE

# Monte Carlo Tree Search (MCTS)



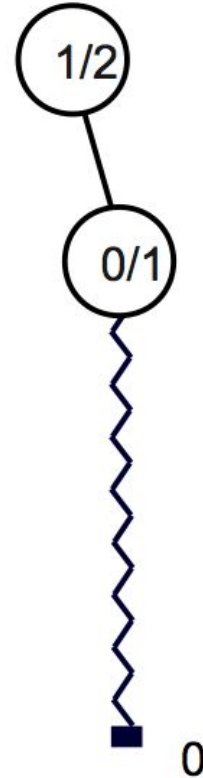
# MCTS: Example

- 1 iteration



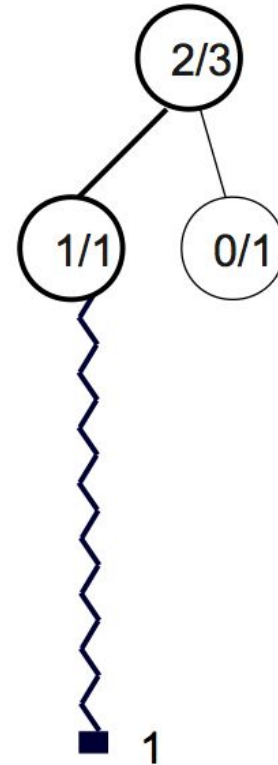
# MCTS: Example

- 2 iterations



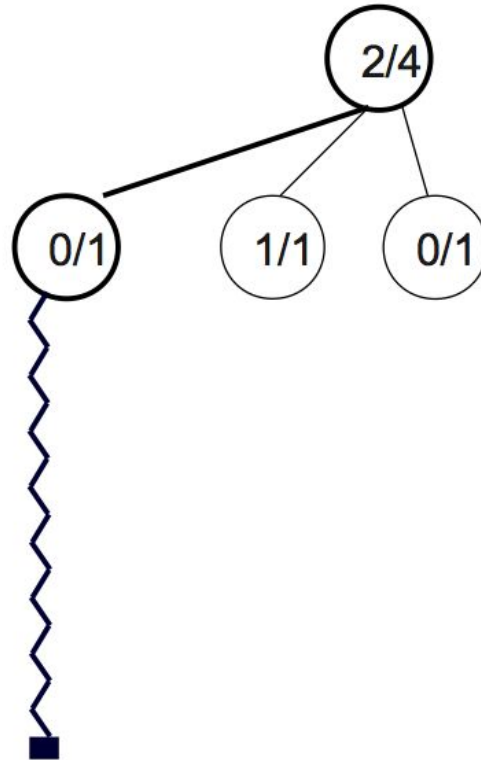
# MCTS: Example

- 3 iterations



# MCTS: Example

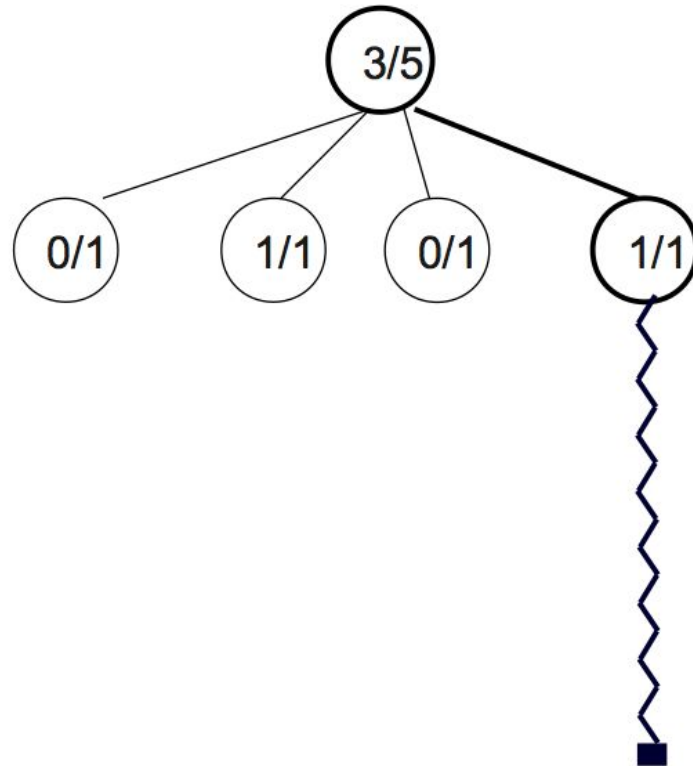
- 4 iterations





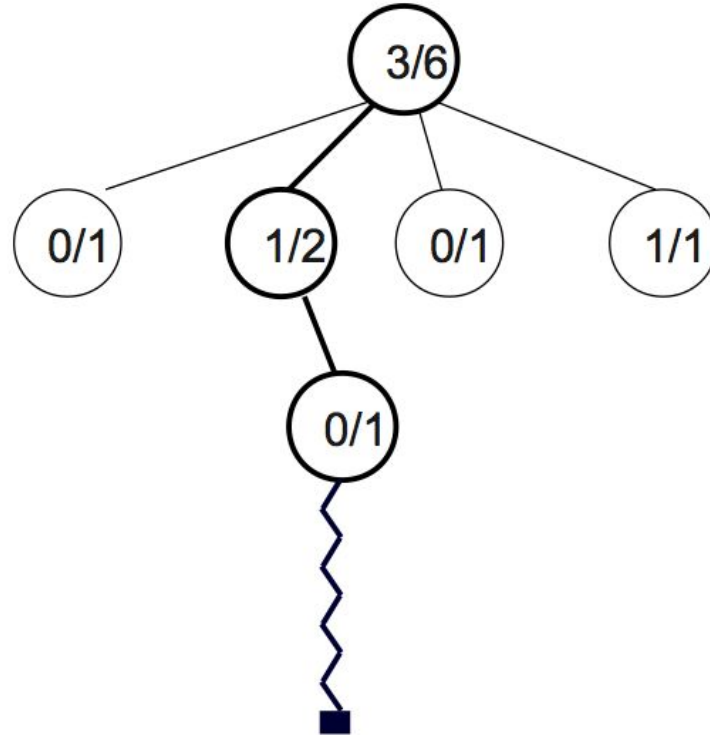
# MCTS: Example

- 5 iterations



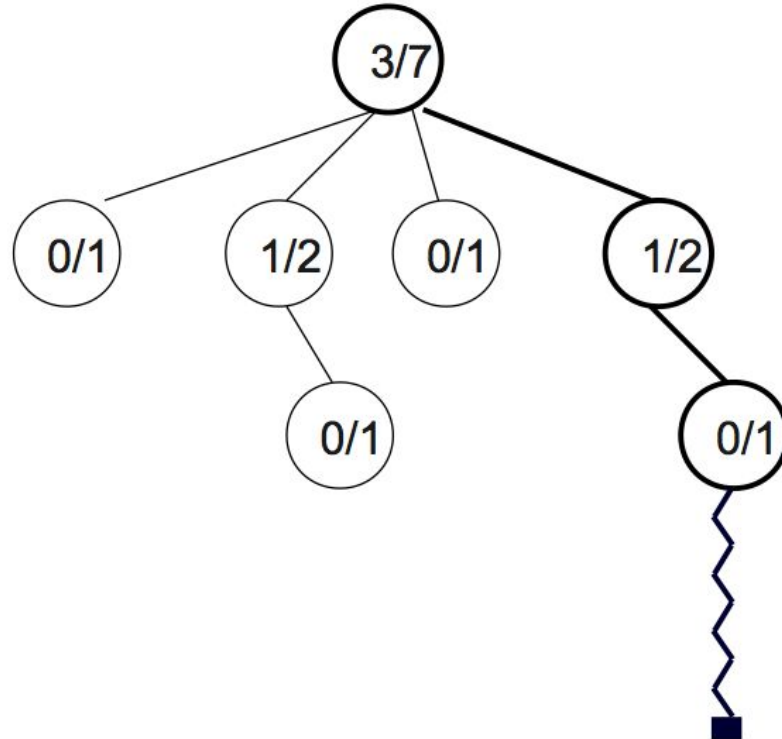
# MCTS: Example

- 6 iterations



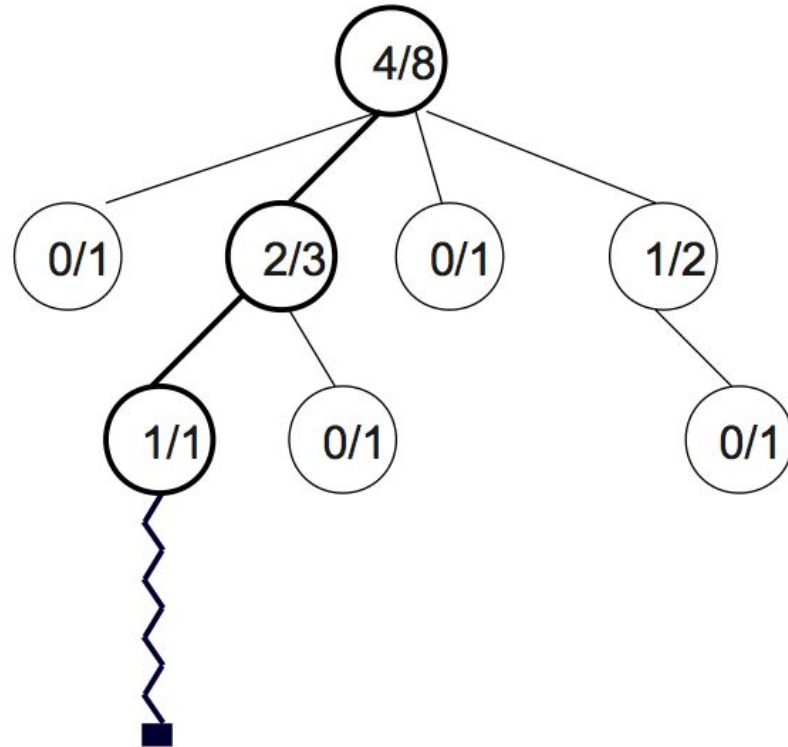
# MCTS: Example

- 7 iterations



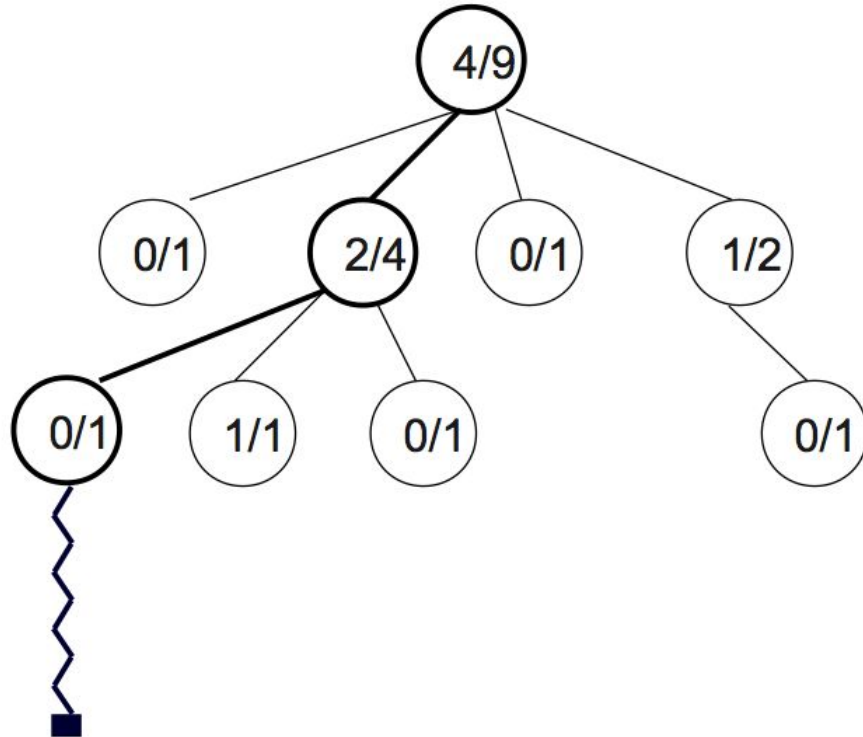
# MCTS: Example

- 8 iterations



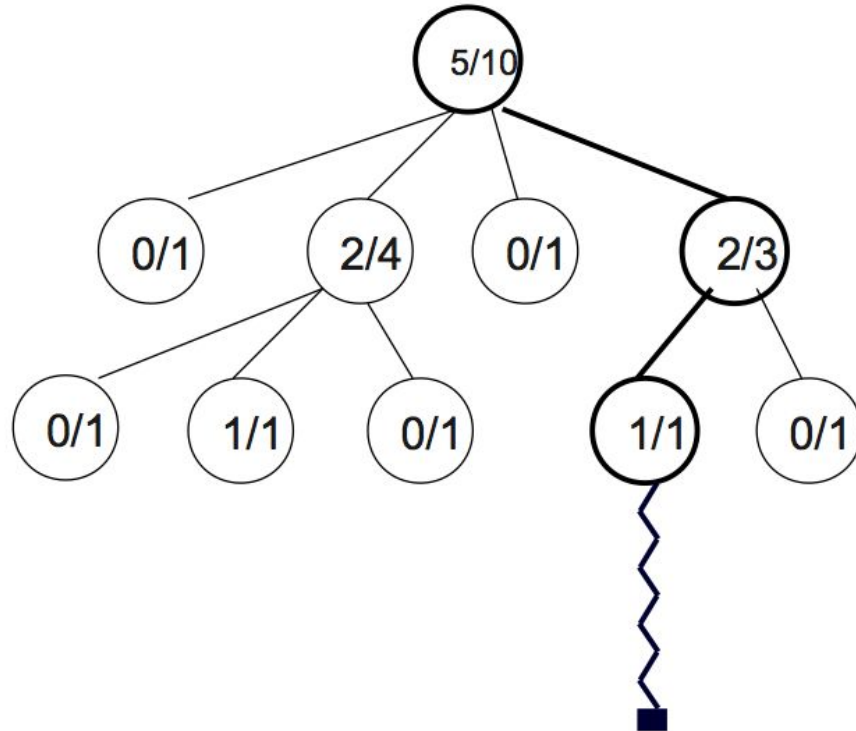
# MCTS: Example

- 9 iterations



# MCTS: Example

- 10 iterations



# AlphaGo

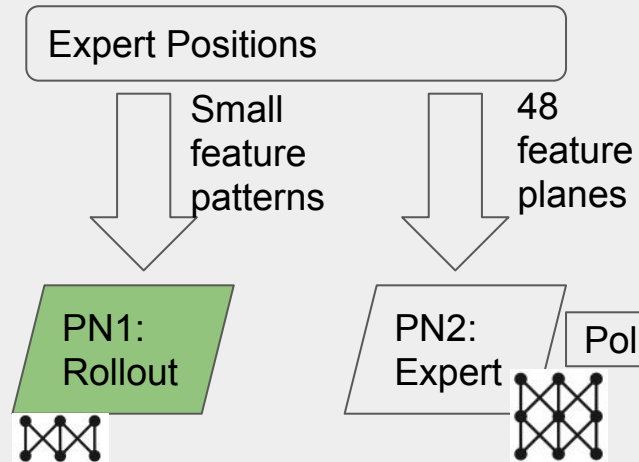
Online Pipeline

Offline

## STEP 1: supervised learning

## S2: reinforcement learning

## STEP3: value networks



Policy gradient

Self-play Positions

PN3: Self



Regression

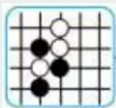
$v_{\theta}(s) \approx v_{\rho}(s) \approx v^*(s)$

Self-play to end

Value Network



Online



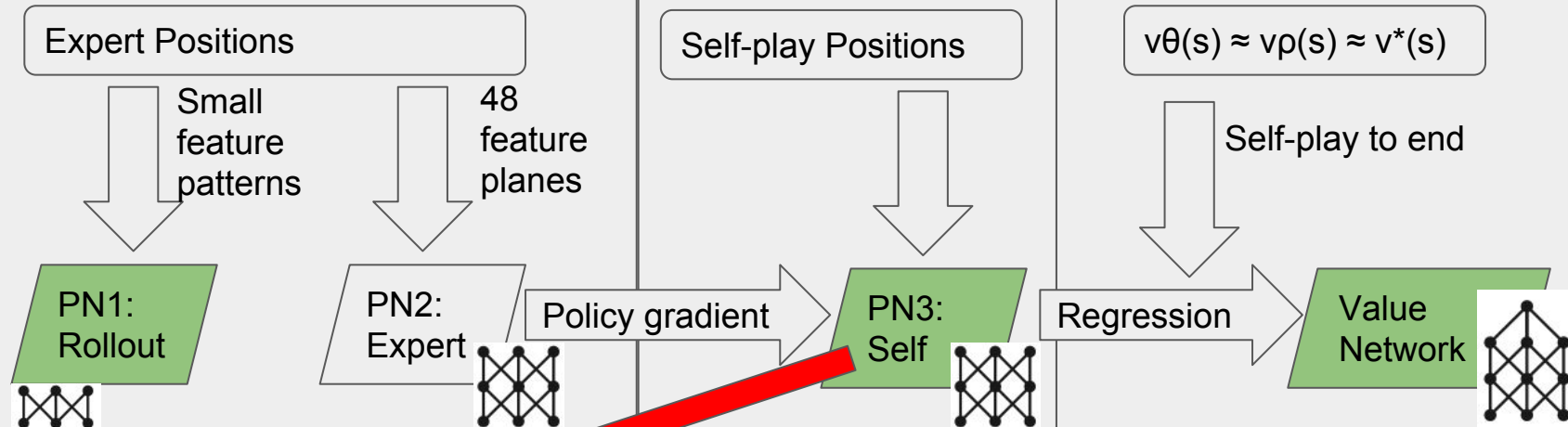


Offline

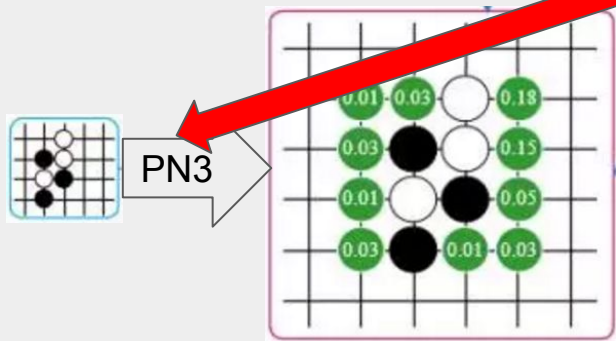
### STEP 1: supervised learning

### S2: reinforcement learning

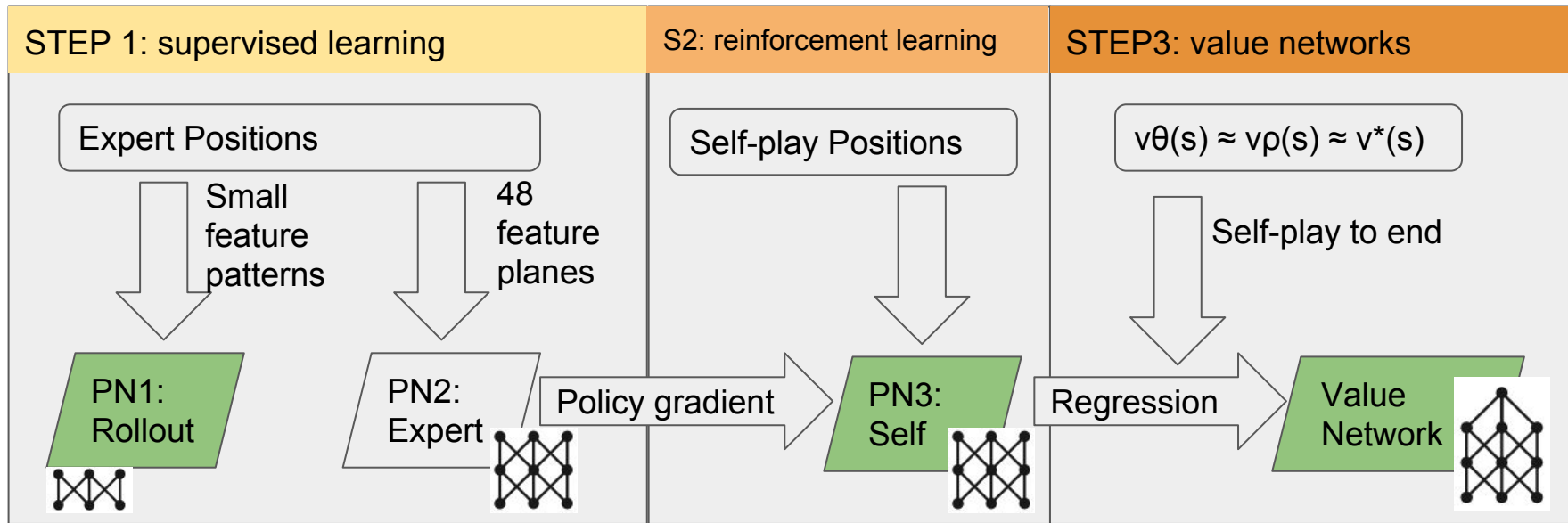
### STEP3: value networks



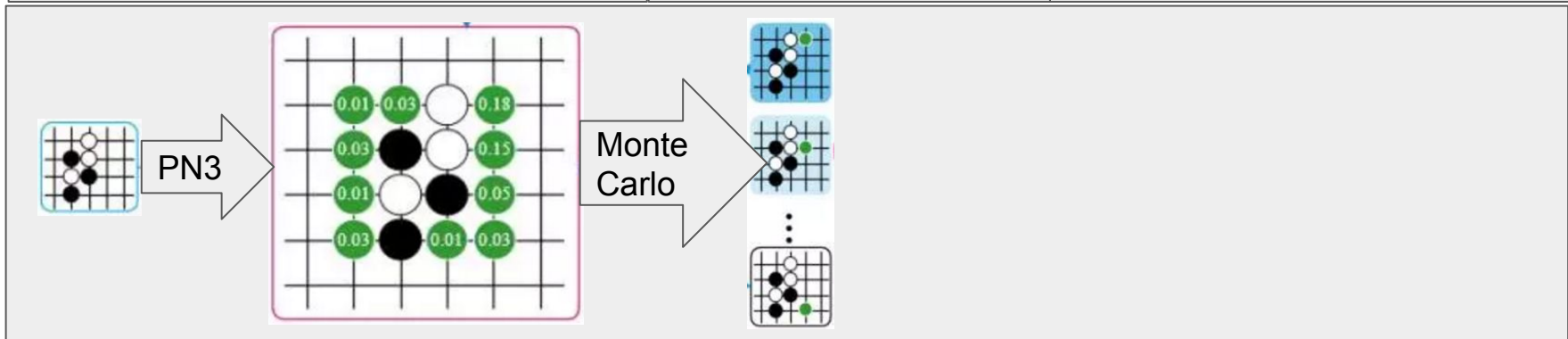
Online



Offline



Online

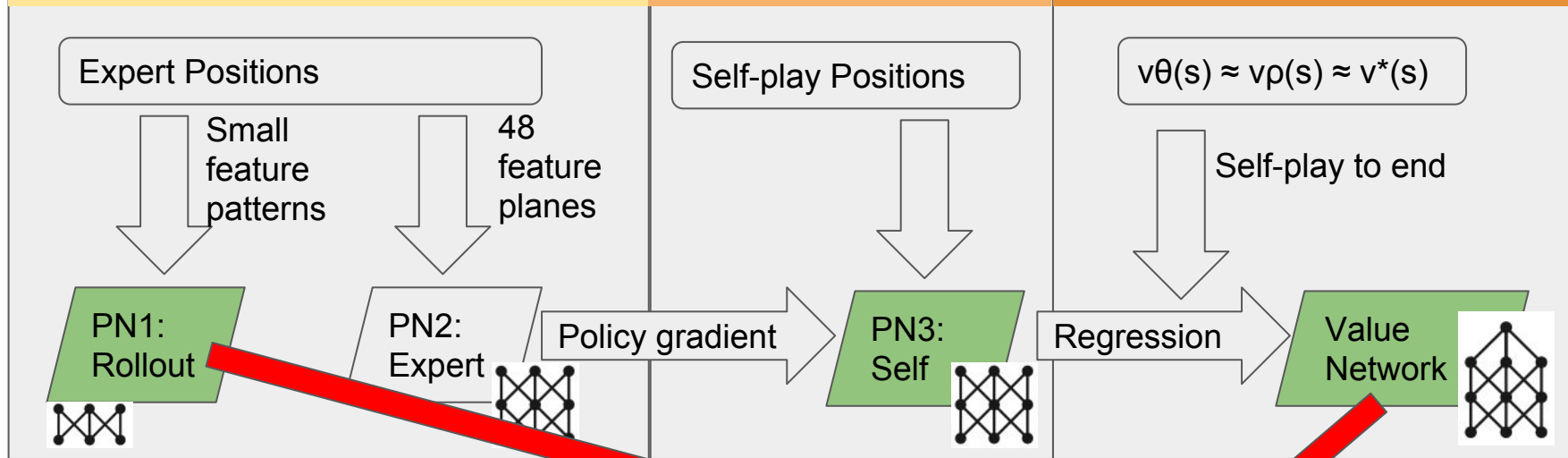


Offline

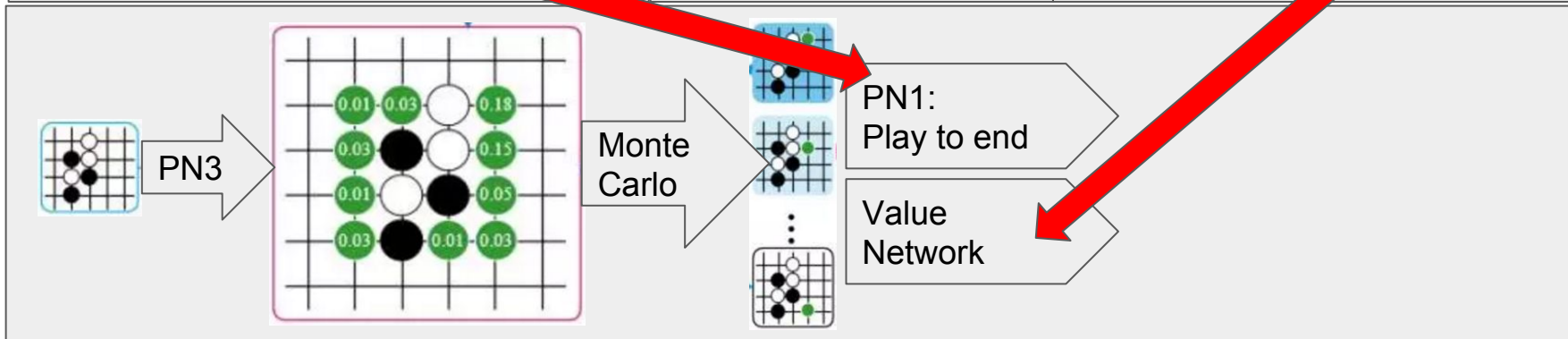
### STEP 1: supervised learning

### S2: reinforcement learning

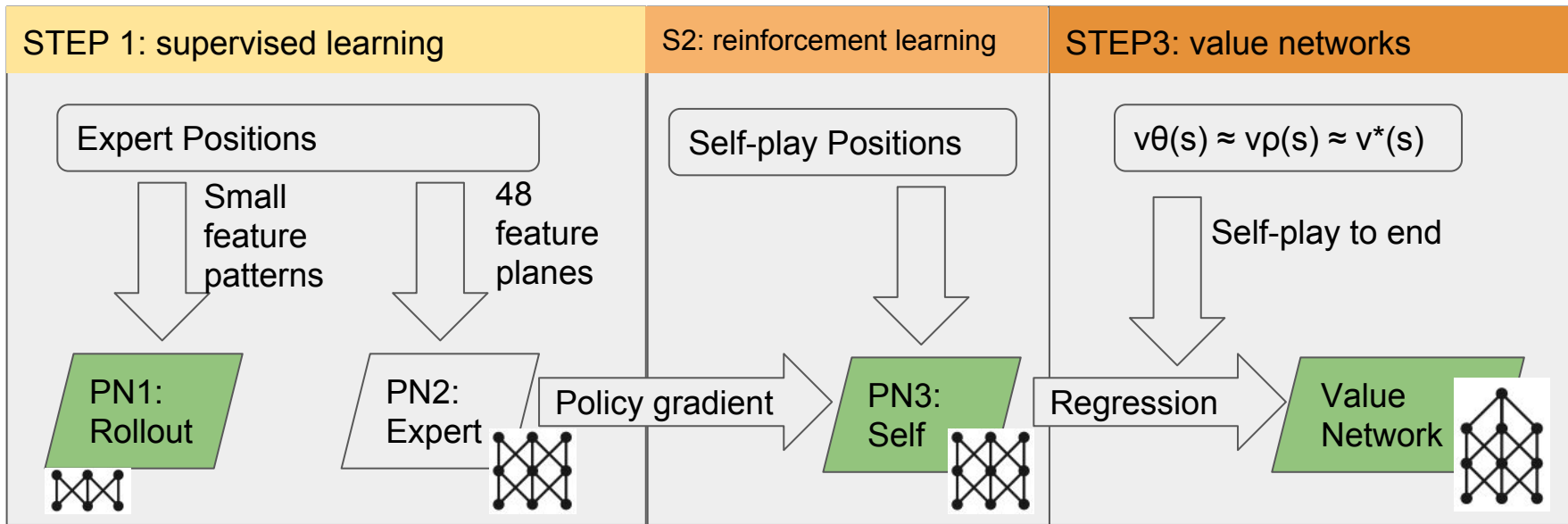
### STEP3: value networks



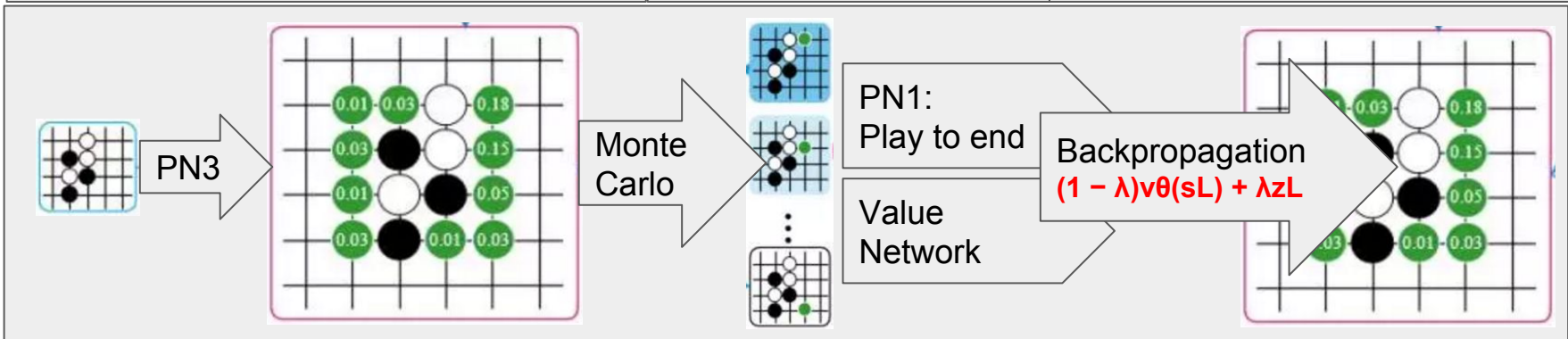
Online



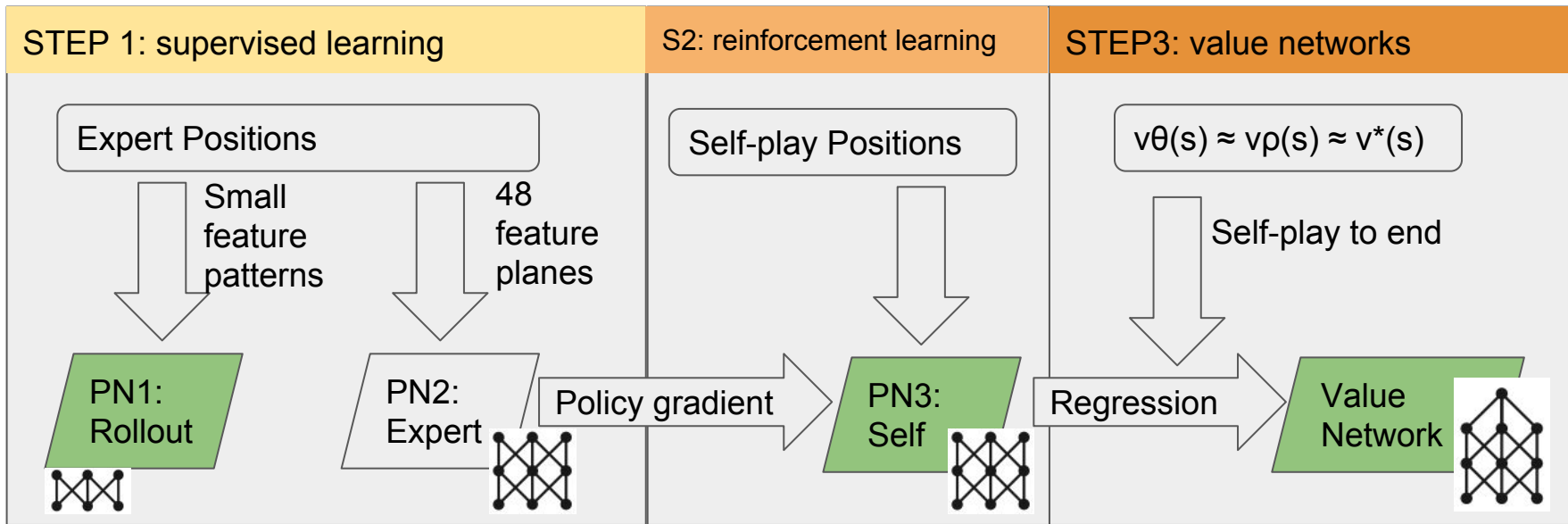
Offline



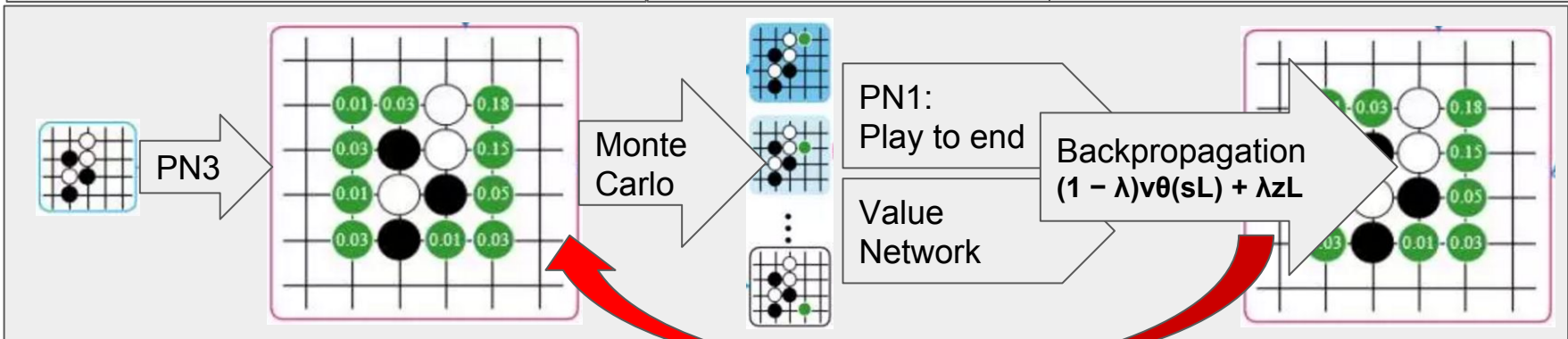
Online



Offline



Online



# Evaluation 2: Won 5-0 to Fan Hui

Neural Network + Monte Carlo

85% => 99.8%

# Thanks

2016/03/15

# Background 3

## Miscellaneous



# Rollout algorithm

- Rollout is a form of sequential optimization that originated in dynamic programming (DP for short). It may be viewed as a single iteration of the fundamental method of policy iteration. The starting point is a given policy (called the base policy), whose performance is evaluated in some way, possibly by simulation. Based on the evaluation, an improved policy is obtained by one-step lookahead.
  - Rollout Algorithms for Discrete Optimization: A Survey, Dimitri P. Bertsekas, MIT

# Monte Carlo Tree Search (MCTS)

- A method for making optimal decisions in artificial intelligence (AI) problems, typically move planning in combinatorial games.
- It combines the generality of random simulation with the precision of tree search.

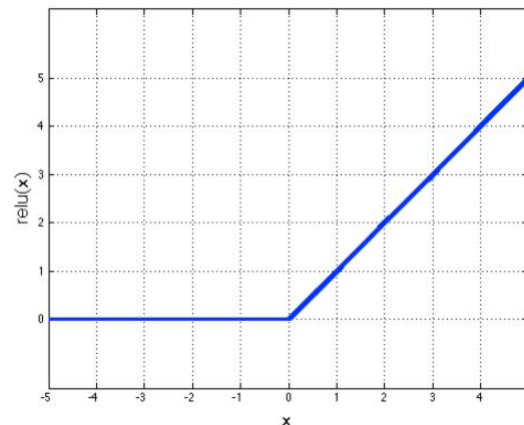
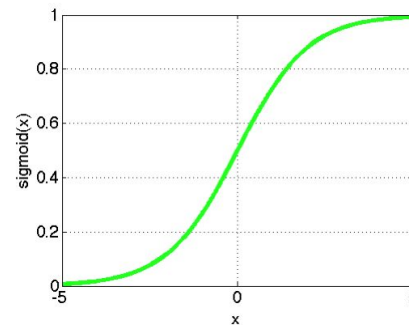
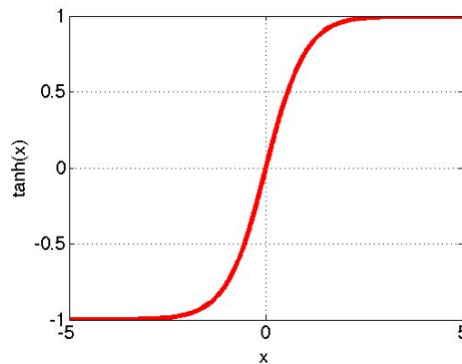


# Game theory

- Types of Games
  - Cooperative or non-cooperative
  - Zero sum and non-zero sum
  - Simultaneous and sequential
  - Perfect information and imperfect information
  - Deterministic vs. stochastic

# Non-linearity: Logistic function

- Rectified linear (ReLU) :  $\max(0, x)$ 
  - Simplifies backprop
  - Makes learning faster
  - Make feature sparse
- Tanh
- Sigmoid:  $1/(1+\exp(-x))$



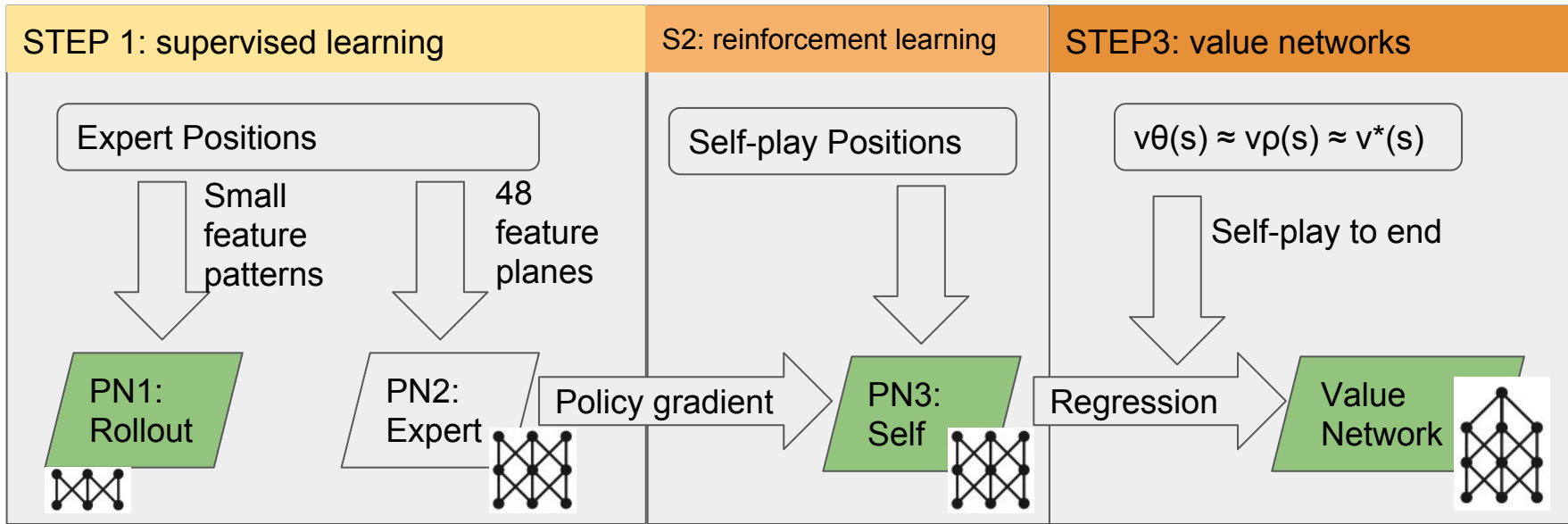
# Elo rating

- A method for calculating the relative skill levels of players in **competitor-vs-competitor** games
- Assumptions
  - players are normally distributed
  - player changes only slowly over time
  - performance can only be inferred from wins, draws and losses

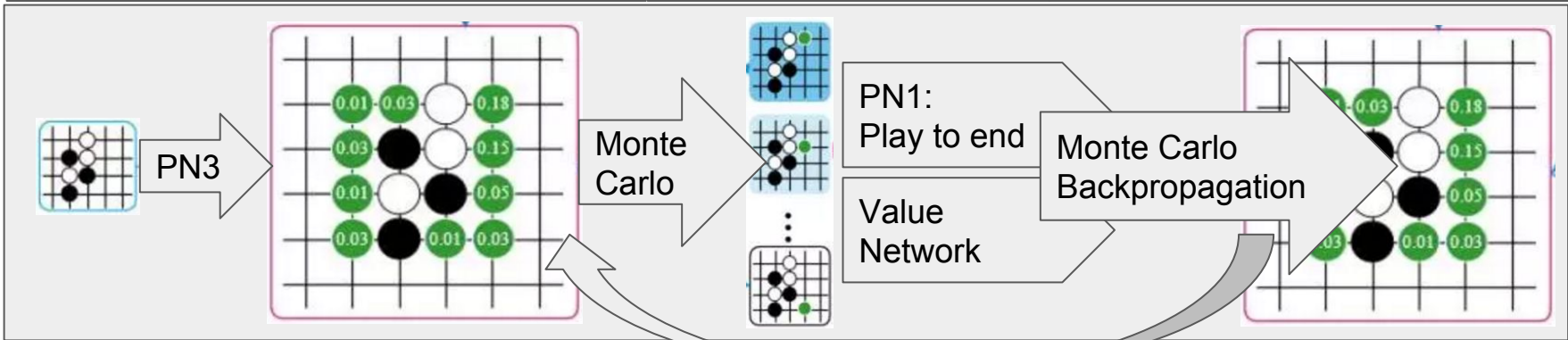


[Arpad Elo](#), the inventor of the Elo rating system

Offline



Online



# Thanks

2016/03/15